第 38 卷第 3 期
录用定稿

计算机应用研究
Application Research of Computers

Vol. 38 No. 3
Accepted Paper



# 基于多尺度特征融合的恶意 HTTP 请求检测方法 *

巫家宏，杨振国，刘文印

(广东工业大学 计算机学院，广州 510006)

**摘　要：**针对当前网络环境中恶意 HTTP 请求攻击泛滥的问题，提出了一种多尺度特征融合的检测方法。首先从单词级和字符级两个尺度对 HTTP 请求进行建模，然后使用卷积神经网络提取其高阶语义特征；再借助多尺度特征融合技术，学习 HTTP 请求的多尺度公共向量表示；最后使用线性分类器进行分类。实验结果表明该方法性能在 HTTP CSIC 2010 数据集和 WAF 真实数据集上优于现有方法。

**关键词：**恶意请求检测；深度学习；特征融合

**中图分类号：**TP391.1　　doi: 10.19734/j.issn.1001-3695.2020.01.0071

## Multiscale feature fusion for malicious HTTP request detection

Wu Jiahong, Yang Zhenguo, Liu Wenyin

(*School of Computer Science*, *Guangdong University of Technology*, *Guangzhou* 510006, *China*)

**Abstract:** This paper proposed a multiscale feature fusion approach for malicious HTTP request detection. Firstly, it modeled the HTTP request in both word-level and character-level. Secondly, it extracted the high level sematic information in HTTP request by using a specially designed convolutional neural network (CNN) . Thirdly, it jointly learnt the multiscale representation for HTTP request with the help of multimodal learning techniques. Finally, it adopted a linear classifier for classification. Experiments on public HTTP CSIC 2010 dataset and WAF dataset show it has large improvement on the performance against existing state-of-the-art methods.

**Key words:** malicious request detection; deep learning; feature fusion

## 0　引言

互联网、移动互联网、物联网的快速发展给人们带来便利的同时，也为个人及国家安全和社会稳定带来新的挑战。当前，针对 Web 系统的攻击手段层出不穷，给广大网络用户和企业造成巨大经济损失，成为社会关注的热点问题之一。如何快速、准确地监测和识别 Web 攻击行为的关键在于理解 HTTP 请求内容，从中识别出恶意 HTTP 请求并且阻止其运行。

常见的恶意请求检测系统和方法可以分为三个类：基于规则，基于统计分析，基于机器学习或深度学习。Denning[1]首先提出基于专家知识的入侵检测系统，这种技术针对已知攻击模式，人工设计、维护规则库和策略模板。对于常见且特征突出的恶意请求来说，基于规则的检测方法准确率较高，但面对新型攻击模式时则表现不佳。Krügel 等人[2]提出了一种检测异常网络流量的统计处理单元，更具体地说，基于请求类型、请求长度、请求内容的分布等三个统计特征来计算请求的分数，系统管理员根据此分数设置阈值来过滤恶意、异常行为。基于统计分析的恶意请求检测技术可以自适应学习用户的行为，但也容易被攻击者绕过，并且不恰当的阈值可能会导致大量漏报、误报产生。汪生等人[3]首先使用模糊 SVM 对原始流量数据进行粗二分类，接着将攻击样本采用 DBscan 模型再次进行细粒度的多类别聚类，从而得到攻击样本的具体类别。Smitha 等人[4]使用一系列机器学习算法，结合最小冗余最大相关性(mRMR)特征选择策略，在 HTTP CSIC 2010 数据集[5]上取得了不错的检测效果。虽然机器学习算法相较于基于规则、基于统计分析的方法有一定的提升，但它们的缺点在于需要借助大量专家知识，手工从 HTTP 请求中提取特征，费时费力。

近年来深度学习技术在计算机视觉、自然语言处理领域成果显著，将深度学习技术应用到恶意请求检测领域是大势所趋。李佳等人[6]提出了直接将 HTTP 请求字符转换为 ASCII 编码作为其向量表示，同时结合若干统计特征，使用具有混合结构的多层神经网络进行分类的方法。Zhang 等人[7]在训练过程中通过 Embedding 层学习 HTTP 请求的单词级向量表示，然后使用 CNN 进行分类检测。Liang 等人[8]首先使用 LSTM 和 GRU 无监督地学习正常 HTTP 请求的单词级特征，然后再训练一个用于分类的多层感知机。Liu 等人[9]提出了基于 HTTP 请求字符级向量表示及 LSTM 的端到端的检测方法。以上研究只使用了 HTTP 请求单一尺度的特征，即对请求只进行单词级建模或者只进行字符级建模，并没有考虑同时使用多个尺度的特征。

受到 Zhen 等人[10]提出的深度有监督跨模态检索框架的启发，本文借鉴多模态学习技术，利用多模态视角来提升分类性能，挖掘 HTTP 请求不同尺度特征之间的互补特性，提出基于多尺度特征融合的恶意 HTTP 请求检测方法。该方法从单词级和字符级两个尺度对 HTTP 请求进行建模，使用 CNN 分别提取单词级和字符级高阶语义特征，然后通过语义特征融合子网络学习请求的多尺度公共向量表示并最终用于线性分类器分类。实验结果表明该方法的分类性能优于现有方法，并且能够学习到有表达力和判别力的 HTTP 请求多尺度公共向量表示。

---

**收稿日期：**2020-01-08；**修回日期：**2020-03-15　**基金项目：**国家自然科学基金资助项目(61703109，91748107)；广东省引进创新科研团队计划资助项目(2014ZT05G157)

**作者简介：**巫家宏(1994-)，男，广东惠州人，硕士研究生，主要研究方向为网络安全、自然语言处理；杨振国(1988-)，男，山东潍坊人，副教授，主要研究方向为自然语言处理、文本挖掘、多媒体；刘文印(1966-)，男，吉林榆树人，教授，硕导，博导，主要研究方向为网络安全、区块链、模式识别等(liuwy@gdut.edu.cn).

## 1　基于多尺度特征融合的恶意 HTTP 请求检测方法

本文将 HTTP 请求样本视为含有一定语义的文本字符串，利用自然语言处理(natural language process，NLP)中文本分类思路，同时结合多模态学习(multimodal learning)技术，提出了一种基于多尺度特征融合的恶意 HTTP 请求检测方法。总体框架如图 1 所示，主要由四个部分组成：a) HTTP 请求文本向量化模块，分别得到 HTTP 请求的单词级与字符级的向量表示；b) 语义特征提取模块，使用 CNN 进一步提取 HTTP 请求向量表示中的高阶语义特征；c) 语义特征融合模块，由多个全连接层组成，将 HTTP 请求的单词级与字符级这两种不同尺度的高阶语义特征映射到一个公共向量空间；d) 分类模块，使用 softmax 线性分类器计算分类结果，即将 HTTP 请求样本分类为恶意或正常。

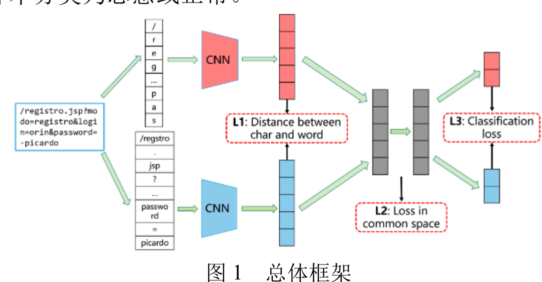



图 1　总体框架

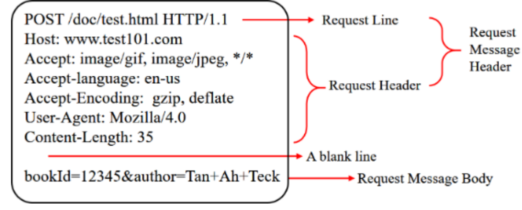
Fig. 1　Overview of the framework

### 1.1　HTTP 请求文本向量化

一个完整的 HTTP 请求由请求头部、空行、请求消息体构成，本文中只提取 HTTP 请求头部中的 URL 路径及请求消息体进行检测，称之为“请求查询字符串(request query string)”，用它来代表整个 HTTP 请求样本。请求查询字符串一般由大小写字母、数字、特殊符号组成，它通常具有这样的键值对结构：

/path?key1=value1&key2=value2&key3=value3...

以图 2 所示的 HTTP 请求为例，本文方法只从中提取出请求查询字符串“*/doc/test.html?bookID=12345&author=Tan+Ah+Teck*”，因为它已经包含了该 HTTP 请求的关键信息。


POST /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck



图 2　HTTP 请求样例

Fig. 2　Example of HTTP request

请求查询字符串中各种字符的使用和出现顺序往往隐含了一定的语义信息，能够直接反映请求发起者的意图，对检测结果起决定性作用。例如，请求查询字符串“*getpage.php?home=../etc/passwd*”是一个典型目录遍历攻击，因为它企图访问服务器中“*etc/passwd*”这一与用户密码存储相关的路径；“*vciar.jsp?B2=Vacar+%27%3B+DROP+TABLE+usuario+SELECT+*+FROM+dato+WHERE+nombre+LIKE* ”则构成了一个 SQL 注入攻击，因为它企图修改服务器数据库，非法获取敏感数据。因此，使用自然语言处理中的神经网络语言模型来无监督地学习 HTTP 请求中的这些语义信息，并将其向量化，方便后续进行进一步高阶特征提取及分类任务。接下来分别介绍 HTTP 请求文本的单词级和字符级向量表示学习方法。

#### 1.1.1 HTTP 请求的单词级向量表示

1)分词

要获得 HTTP 请求的单词级向量表示，首先要将 HTTP 请求进行单词级的分词处理。请求查询字符串一般由若干键值对组成，每个键值对之间由符号“&”分隔，而每个键值对的键与值之间由符号“=”连接。因此，可以以“&”、“=”及其他特殊符号为分隔符，将请求查询字符串划分为若干单词和符号序列。例如，图 2 中的请求样例经单词级分词处理后得到序列(“doc”，“test”，“html”，“bookID”，“12345”，“author”，“Tan”，“Ah”，“Teck”)。将数据集中所有请求样例都进行单词级分词处理后，得到此数据集的单词级语料库。

2)向量化

传统的文本向量化方法如 One-hot 编码，可以将每个单词表示为只有一个元素为 1，其余元素为 0 的稀疏向量。One-hot 编码计算简单，一定程度上对特征进行了扩充；然而，它并没有考虑单词与单词之间的联系，任意两个单词之间是孤立的，无法表达不同单词组合所蕴涵的语义信息。当语料库规模很大时还会产生维度爆炸问题。word2vec[11]是当前自然语言处理领域较为常用且有效的文本向量化技术。word2vec 基于分布假设理论，不同的单词如果出现在相同的上下文环境中会有相似的语义，即具有相似语义的单词的向量表示之间的距离比较小，单词之间的关系可以用其向量表示的运算来表达。

HTTP 请求文本与普通文本字符串类似，其中蕴涵的语义信息同样可以使用 word2vec 来捕获。word2vec 中的 skip-gram 模型用目标单词预测其上下文单词出现的概率，训练得到的词向量与其他语言模型相比更加准确，因此，在恶意 HTTP 请求检测这一任务中本文采用 Skip-gram 模型来学习 HTTP 请求的向量表示。

具体地，首先以步骤 1)构造的单词级语料库作为输入，使用 skip-gram 模型训练出一个单词级的语言模型；对于 HTTP 请求样本中的每个单词，找到其在这个单词级语言模型中对应的单词级词嵌入(embedding)；将 HTTP 请求样本中的每个单词的单词级词嵌入纵向堆叠起来，则得到整个 HTTP 请求的单词级二维向量表示。

#### 1.1.2 HTTP 请求的字符级向量表示

1)分词

要获得 HTTP 请求文本的字符级向量表示，首先要将 HTTP 请求进行字符级的分词处理。与单词级分词不同，字符级分词只需要简单地将请求查询字符串将划分为单个字符组成的序列。例如，图 2 中的 HTTP 请求样例经字符级分词处理后得到序列(“d”，“o”，“c”，“t”，“e”，…，“A”，“h”，“T”，“e”，“c”，“k”)。将数据集中所有请求样例都进行字符级分词处理后，得到此数据集的字符级语料库。

2)向量化

HTTP 请求的字符级向量化与单词级向量化类似。首先以字符级语料库作为输入，训练出一个字符级的语言模型；对于 HTTP 请求样本中的每个字符，找到其在这个字符级语言模型中的对应字符的字符级词嵌入；将请求查询字符串中的每个字符的字符级词嵌入纵向堆叠起来，则得到整个 HTTP 请求的字符级二维向量表示。

### 1.2　语义特征提取

CNN 是大多数计算机视觉系统的核心技术，它通过使用多个不同尺度的卷积核来提取图像像素之间的局部相关性。近些年来，CNN 在自然语言处理领域也作出了巨大贡献。恶意 HTTP 请求检测也可以被视为自然语言处理中一种特殊的文本分类任务。HTTP 请求的单词级和字符级二维向量表示与图像的像素矩阵类似，受文献[12]的启发，本文构建了特殊的 CNN 模型，用来进一步从中提取高阶语义特征。其模型结构如图 3 所示，它主要包含 4 个卷积层(当模型输入为单词级向量表示时，有 3 个卷积层)和 1 个全连接层。

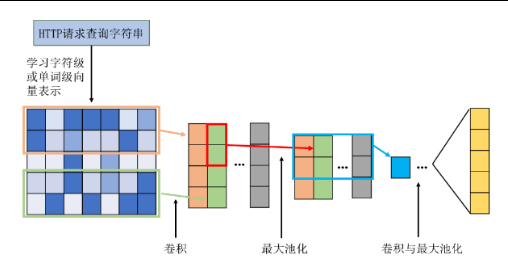



图 3　语义特征提取子网络结构

Fig. 3　Architecture of sematic feature extraction network

语义特征提取子网络的输入是 HTTP 请求的单词级或字符级二维向量表示，对每个卷积层的输出都使用 Relu 激活函数激活，并进行最大池化，最后的全连接层输出每个 HTTP 请求对应的一维的原始高阶语义特征向量。值得注意的是，第一个卷积层的卷积核宽度与 HTTP 请求中各个字符或单词的向量表示的维度一致，同时，每个卷积层的卷积核只在请求文本的二维向量表示的垂直方向滑动遍历。这样做的目的是保证 CNN 进行卷积操作时的最小粒度为单词级或字符级，并且能捕获多个连续单词或连续字符之间蕴涵的类似“N-gram”特征的语义信息。此外，宽通道可以让每个卷积层学习到更加丰富的特征，增强模型的表达能力。

## 1.3　语义特征融合

HTTP 请求的单词级向量表示和字符级向量表示从不同角度和颗粒度描述同一个 HTTP 请求样本，它们分布在不同的向量空间，具有不同的统计属性。如何同时充分利用 HTTP 请求两种不同尺度向量表示中的有效语义特征是提高分类性能的关键。常见的特征级融合方法如将多个特征向量串行拼接或直接相加虽然操作简单高效，但作用于不同尺度的特征向量时会带来信息丢失。借鉴多模态学习的思想，利用 HTTP 请求单词级和字符级向量表示的互补性，去除它们之间的冗余性，可以学习到更有表达力和判别力的多尺度向量表示，获得更好的检测结果。

本文提出的语义特征融合子网络由两个全连接层组成，目的是使 1.2 节中学习到的属于不同尺度的单词级与字符级高阶语义特征向量相互作用，并映射到一个公共向量子空间，使得这两种语义特征向量的分布尽可能趋于一致。具体地，将这两种不同尺度的语义特征先后输入至全连接层，强迫它们共享全连接层的权值，结合相关损失函数，最小化属于相同类别(恶意或正常)但不同尺度的语义特征向量之间的距离，同时最大化属于不同类别语义特征向量之间的距离。

## 1.4　目标函数

本文提出的基于多尺度特征融合的恶意 HTTP 请求检测方法在模型训练阶段有三个训练目标：a) HTTP 请求的单词级与字符级高阶语义特征；b) 一个使得来自不同尺度但相同类别的 HTTP 请求的向量表示更接近的公共向量子空间；c) HTTP 请求的类别标签。本文使用三个不同损失函数对它们进行联合优化学习。

### 1.4.1 多尺度不变性损失

为了保证多尺度特征的不变性，需要最小化来自不同尺度但相同类别的所有 HTTP 请求样本的向量表示之间的距离。具体地，本文使用 MMD(Maximum Mean Discrepancy，最大均值差异)来最小化 HTTP 请求的单词级与字符级语义特征分布之间的距离，以消除这两种尺度特征之间的差异。MMD 是迁移学习，尤其是域适应(Domain adaption)中广泛使用的损失函数，它度量在再生希尔伯特空间中两个向量分布的距离。因此，HTTP 请求的单词级和字符级的高阶语义特征的 MMD 多尺度间不变性损失$\mathcal{L}_1$可以定义为

$$\mathcal{L}_1 = \sum_{i=1}^{n}\phi(W_i) - \sum_{i=1}^{n}\phi(C_i)_H^2 \tag{1}$$

其中 $W$ 为 HTTP 请求的单词级语义特征向量，$C$ 为字符级语义特征向量，$\phi(\cdot)$ 为映射函数，用于将原始向量映射到再生希尔伯特空间中。

### 1.4.2 公共空间损失

为了使得在公共向量子空间中，相同类别的 HTTP 请求的向量表示更加接近，而不同类别 HTTP 请求的向量表示更有区分度，参考文献[10]，本文使用公共空间损失 $\mathcal{L}_2$ 直接度量在公共向量子空间两种不同尺度的所有请求样本的判别损失：

$$\mathcal{L}_2 = \frac{1}{n^2}\sum_{i,j=1}^{n}\left(\log\left(1+e^{\Gamma_{i,j}}\right) - S_{ij}^{wc}\Gamma_{i,j}\right) + \frac{1}{n^2}\sum_{i,j=1}^{n}\left(\log\left(1+e^{\phi_{i,j}}\right) - S_{ij}^{ww}\phi_{i,j}\right) + \frac{1}{n^2}\sum_{i,j=1}^{n}\left(\log\left(1+e^{\theta_{i,j}}\right) - S_{ij}^{cc}\theta_{i,j}\right) \tag{2}$$

其中 $\Gamma_{i,j} = 1/2\cos\left(w_i, c_j\right)$， $\phi_{i,j} = 1/2\cos\left(w_i, w_j\right)$， $\theta_{i,j} = 1/2\cos\left(c_i, c_j\right)$， $w$ 和 $c$ 分别是单词级和字符级 HTTP 请求在公共子空间中对应的向量表示； $\cos(?)$ 计算两个向量之间的余弦相似度； $S_{ij}^{uv}$ 是一个指示函数，当 $u$ 和 $v$ 为类别相同的公共子空间向量表示时值为 1，否则为 0。式(2)中第一项用来衡量不同尺度的公共向量表示的相似度，第二项衡量所有单词级公共向量表示的相似度，第三项衡量所有字符级公共向量表示的相似度。因此，公共空间损失 $\mathcal{L}_2$ 可以合理地计算 HTTP 请求的单词级和字符级语义特征经公共向量子空间映射得到的公共向量表示的相似度。

### 1.4.3 分类损失

本文使用分类问题中常用的交叉熵损失来衡量最终的分类效果，可以定义为

$$\mathcal{L}_3 = CrossEntropy\left(y, \hat{p}_w\right) + CrossEntropy\left(y, \hat{p}_c\right) \tag{3}$$

其中 $CrossEntropy(\cdot)$ 为交叉熵损失函数，$y$ 为样本真正的标签，$p_w$ 为单词级公共向量表示经线性分类器计算后预测的样本标签，$p_c$ 字符级公共向量表示预测的样本标签。

结合公式(1)~(3)，得到了最终的目标函数：

$$\mathcal{L} = \lambda\mathcal{L}_1 + \eta\mathcal{L}_2 + \mathcal{L}_3 \tag{4}$$

其中超参数 $\lambda$ 和 $\eta$ 为损失权重系数。最后使用 Adam 算法优化公式(4)。

# 2　实验结果与分析

在本章从以下三个角度对本文提出的基于多尺度特征融合的恶意 HTTP 请求检测方法进行评测：1)本方法的能否取得令人满意的性能表现，特别是在真实网络环境下？2)本方法中训练过程使用的三个损失函数的有效性如何？3)本方法能否从 HTTP 请求学习到有判别力的特征及向量表示？

## 2.1　数据集

### 2.1.1 HTTP CSIC 2010 数据集

HTTP CSIC 2010 数据集[5]是由西班牙研究委员会(CSIC)信息安全研究所编制，由 36000 条正常 HTTP 请求样本和 25000 条恶意 HTTP 请求样本组成。其中，它包含如 SQL 注入、文件披露、信息收集、跨站脚本、参数窜改、非法用户行为等多种攻击类型。它主要用于测试网络攻击保护系统，是恶意请求检测领域被广泛使用的通用数据集。

### 2.1.2 WAF 真实数据集

为了进一步验证本文提出的方法在现实场景中的有效性和通用性，本文还在 WAF 真实数据集[13]进行测试。它由网络安全研究人员从真实网络流量中收集并发布在 Github 上。经过数据清洗和预处理后从中随机选择 50000 条正常 HTTP 请求样本和 46938 恶意 HTTP 请求样本构造数据集。

## 2.2　评价指标

本文使用了异常检测领域中常用的几个评价指标进行量化分析，分别是：误报率(False positive rate，FPR)，召回率

(true positive rate，TPR)，准确率(accuracy，ACC)。

## 2.3 对比方法

本文将所提出方法分别与当前领域中较新且效果较好的方法进行对比分析。具体描述如下：

a) SVM/LR[4]。基于 SVM/LR 的 Web 入侵检测方法，使用 mRMR 进行特征提取。

b) C4.5[14]。基于词频的决策树模型，并且使用了 HTTP 请求中的长度、特殊表达式作为特征。

c) RE[15]。正则匹配异常检测模型，使用图分割和动态规划技术来获取正则表达式。

d) FSWAF[16]。一个 WAF 系统，基于关联性分析和 mRMR 进行特征选择，并使用多个机器学习分类器进行分类。

e) Word-CNN[7]。单词级 CNN 模型，使用 Embedding 层随机初始化 HTTP 请求的单词级向量表示，并在训练过程中更新。

f) AB-LSTM/AB-GRU[8]。基于 LSTM/GRU，首先只使用正常 HTTP 请求进行无监督训练，然后再用 LSTM/GRU 的输出训练一个分类器。

g) PL-LSTM [9]。端到端的基于 LSTM 的恶意请求检测方法，使用了预训练的字符级向量表示。

## 2.4 HTTP CSIC 2010 数据集结果与对比

本方法及基线方法在 HTTP CSIC 2010 数据集上的实验结果如表 1 所示。由于这些基线方法没有公布源代码，也不知道它们训练与测试时的数据划分情况，因此，本文统一汇报这些基线方法在其原始论文中的最佳结果。从表 1 可以看出，前五个基于机器学习的方法取得比较接近的效果；而基于词嵌入的深度学习方法，如 PL-LSTM，AB-LSTM，AB-GRU 及本方法性能均优于机器学习方法，这说明了词嵌入能够从 HTTP 请求中捕获有效的语义信息。在几个基于深度学习的方法中，Word-CNN 的性能明显差于其他方法，主要原因是因为它使用的 HTTP 请求的向量表示是随机生成的，这其中并没有包含关于 HTTP 请求的先验知识。AB-LSTM 和 AB-GRU 由于使用了预训练的单词级请求词嵌入，并且在训练过程对其进行优化更新，因此性能优于在训练过程中固定权重的 PL-LSTM。

表 1　在 HTTP CSIC 2010 数据集的结果

Tab. 1　Performance on HTTP CSIC 2010 dataset

| Method | FPR | TPR | ACC |
|---|---|---|---|
| C4.5 | - | 96.3 | 96.26 |
| SVM | - | 95 | 97 |
| LR | - | 97 | 92 |
| FSWAF | 2.15 | 98.02 | - |
| RE | 4.34 | 94.46 | - |
| Word-CNN | 1.37 | 93.35 | 96.49 |
| PL-LSTM | - | 97.79 | 96.13 |
| AB-LSTM | 0.79 | 97.56 | 98.42 |
| AB-GRU | 1.55 | 97.2 | 97.88 |
| **本文方法** | **0.20** | **98.65** | **99.34** |

本文提出的方法在不依赖专家知识和不需要人工提取特征的情况下，性能优于所有基线方法，准确率达到 99.34%，召回率达到 98.65%，而误报率只有 0.2%。本文方法之所以有效主要有两个方面的原因：1)使用 Skip-gram 语言模型从单词级和字符级两个尺度对 HTTP 请求进行建模，能够有效地保留 HTTP 请求中隐含的丰富的语义信息；2)构造的语义特征提取子网络和语义特征融合子网络能够有效地提取 HTTP 请求的单词级和字符级高阶语义特征，并且通过多尺度特征融合的方式得到更有表达力和判别力的 HTTP 请求公共向量表示。

## 2.5 消融实验

由于 HTTP CSIC 2010 数据集是人工构造出来的，同时包含一些拉丁字符，与当前网络环境中的攻击流量有一定差距。为了验证本方法在现实网络环境中的表现，余下实验评测都在 WAF 真实流量数据集上进行。

### 2.5.1 多尺度公共向量表示有效性分析

本文从单词级和字符级两个尺度对 HTTP 请求进行建模，并且在训练过程中对这两个尺度的语义特征进行联合学习，得到 HTTP 请求在公共向量子空间的多尺度公共向量表示。为了验证其有效性，本实验中将模型分解，构造了三个模型变体：W-CNN(只使用单词级请求向量表示)、C-CNN(只使用字符级请求向量表示)、WC-CNN(将单词级和字符级请求向量表示直接相加)，与本文方法在 WAF 真实数据集进行对比。由表 2 所示的消融实验结果可以看出：W-CNN 性能最差，可能是由于以“&”、“=”等符号作为分隔符的分词的策略比较粗糙，对 HTTP 请求文本进行了不合理的切割，导致学习到的单词级向量表示不够准确；W-CNN 和 C-CNN 这两个只利用了单一尺度语义信息的模型总体来说性能不如其他两个利用多个尺度语义信息的模型；WC-CNN 将 HTTP 请求的两种尺度的向量表示简单相加，综合一部分跨尺度语义信息，但不同尺度特征之间没有产生交互，丢失了一部分各个尺度特有的语义信息。而本文方法同时从单词级和字符级两个尺度对 HTTP 请求进行建模，并且在训练过程中将两个尺度特征进行了有效的交互和融合，学习到 HTTP 请求的多尺度公共向量表示，在 WAF 真实数据集取得了最好的分类性能：误报率只有 0.03%，召回率达到 99.92%，准确率达到 99.95%。这样的表现也充分说明了本文方法对复杂的真实网络攻击有良好的泛用性和鲁棒性，可以在真实生产环境中发挥重要作用。

表 2　不同尺度向量表示在 WAF 真实数据集的结果

Tab. 2　Performance of different modal representation on WAF dataset

| Method | FPR | TPR | ACC |
|---|---|---|---|
| W-CNN | 0.48 | 97.44 | 98.52 |
| C-CNN | 0.06 | 99.92 | 99.91 |
| WC-CNN | 0.07 | 99.93 | 99.93 |
| **本文方法** | **0.03** | **99.94** | **99.95** |

### 2.5.2 损失函数有效性分析

本文在训练过程中使用了三个损失函数对模型进行联合优化学习，使得不同尺度但相同类别的 HTTP 请求的向量表示更加接近。本节进一步评测和分析各个损失函数对模型性能的影响，构造了本文方法的变体：V1(不使用多尺度不变损失 L1)、V2(不使用公共空间损失 L2)，及 V3(使用全部损失函数)。它们在 WAF 数据集上表现如表 3 所示。可以观察到，当使用全部三个损失函数时取得最好的性能，这意味着它们都对最终的检测结果有贡献。V3 各个指标都领先 V1，证明多尺度不变性损失 L1 能够有效消除不同尺度特征间的差异。V2 稍差于 V3，证明公共空间损失 L2 对于融合多尺度语义特征的重要性。以上结果和分析表明，这三个损失函数对于融合不同尺度 HTTP 请求的向量表示，提高分类性能确实有效。

表 3　各损失函数在 WAF 真实数据集的结果

Tab. 3　Performance comparison of loss functions on WAF dataset

| Method | FPR | TPR | ACC |
|---|---|---|---|
| V1 | 0.05 | 99.91 | 99.93 |
| V2 | 0.04 | 99.93 | 99.94 |
| V3 | **0.03** | **99.94** | **99.95** |

## 2.6 多尺度公共向量表示可视化

为了直观地观察本文方法学习到的 HTTP 请求的多尺度公共向量表示是否具有区分度，本节使用 t-SNE[17]将 WAF 真实数据集的测试集样本映射到二维向量空间并可视化，如图

4 所示。可以观察到，恶意请求样本与正常请求样本明显地区分开来，绝大部分恶意请求样本分布在图的中部，正常请求样本分布在图的四周；只有少数几个恶意请求样本与正常请求样本重叠。这充分说明本文提出方法能够学习到具有良好的表达力和判别力的 HTTP 请求多尺度公共向量表示，进而提高分类性能。

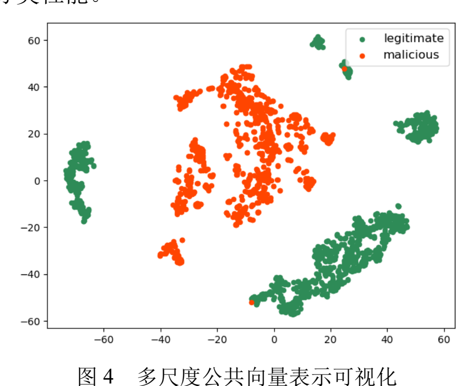



图 4　多尺度公共向量表示可视化

Fig. 4　Visualization of multiscale representation

## 3　结束语

本文提出了基于多尺度特征融合的恶意 HTTP 请求检测方法，将 HTTP 请求视作具有一定语义的字符串，从单词级和字符级两个尺度对 HTTP 请求文本进行建模，使用 CNN 提取高阶语义特征后，再通过多尺度特征融合技术学习其多尺度公共向量表示，最终用于分类检测。多个对比实验表明，本文提出方法在 HTTP CSIC 2010 数据集和 WAF 真实数据集上表现优秀，同时能够学习到有判别力和表达力的 HTTP 请求向量表示。下一步工作将继续探讨更有效的多尺度特征融合方法，寻找性能更好的分类算法。

## 参考文献：


[1] Denning D E. An intrusion-detection model [J]. IEEE Trans on Software Engineering, 1987 (2): 222-232.

[2] Krügel C, Toth T, Kirda E. Service specific anomaly detection for network intrusion detection [C]// Proc of the ACM Symposium on Applied Computing. ACM, 2002: 201-208.

[3] 汪生，金志刚. 基于模糊 SVM 模型的入侵检测分类算法 [J/OL]. 计算机应用研究，2018, 37 (2) . (Wang Sheng, Jin Zhigang. IDS classification algorithm based on fuzzy SVM models [J/OL]. Application Research of Computers, 2018, 37 (2) .)

[4] Smitha R, Hareesha K S, Kundapur P P. A machine learning approach for web intrusion detection: MAMLS perspective [C]// Soft Computing and Signal Processing. Springer, Singapore, 2019: 119-133.

[5] HTTP DATASET CSIC 2010, http://www. isi. csic. es/dataset/.

[6] 李佳，云晓春，李书豪，等. 基于混合结构深度神经网络的 HTTP 恶意流量检测方法 [J]. 通信学报，2019, 40 (01): 24-33. (Li Jia, Yun Xiaochun, Li Shuhao, *et al.* HTTP malicious traffic detection method based on hybrid structure deep neural network [J]. Journal on Communications, 2019, 40 (01): 24-33.)

[7] Zhang Ming, Xu Boyi, Bai Shuai, *et al.* A deep learning method to detect web attacks using a specially designed CNN [C]// Proc of International Conference on Neural Information Processing. Springer, Cham, 2017: 828-836.

[8] Liang Jingxi, Zhao Wen, Ye Wei. Anomaly-based web attack detection: a deep learning approach [C]// Proc of the International Conference on Network, Communication and Computing. ACM, 2017: 80-85.

[9] Liu Hongyu, Lang Bo, Liu Ming, *et al.* CNN and RNN based payload classification methods for attack detection [J]. Knowledge-Based Systems, 2019, 163: 332-341.

[10] Zhen Liangli, Hu Peng, Wang Xu, *et al.* Deep supervised cross-modal retrieval [C]// Proc of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 10394-10403, 2019.

[11] Mikolov T, Sutskever I, Chen K, *et al.* Distributed representations of words and phrases and their compositionality [C]// Proc of Neural Information Processing Systems. 2013: 3111-311.

[12] Zhang Xiang, Zhao Junbo, LeCun Y. Character-level convolutional networks for text classification [C]// Proc of Neural Information Processing Systems. 2015: 649-657.

[13] WAF dataset, https://github. com/faizann24/Using-machine-learning-to-detect-malicious-URLs.

[14] Sahin M, Sogukpınar I. An efficient firewall for web applications (EFWA) [C]// Proc of International Conference on Computer Science and Engineering. IEEE, 2017: 1150-1155.

[15] Choraś M, Kozik R. Machine learning techniques applied to detect cyber attacks on web applications [J]. Logic Journal of the IGPL, 2015, 23 (1): 45-56.

[16] Torrano-Gimenez C, Nguyen H T, Alvarez G, *et al.* Applying feature selection to payload-based web application firewalls [C]// Proc of the 3rd International Workshop on Security and Communication Networks (IWSCN) . IEEE, 2011: 75-81.

[17] Maaten L, Hinton G. Visualizing data using t-SNE [J]. Journal of Machine Learning Research, 2008, 9 (Nov): 2579-2605.